



WEB SCRAPPER

NewsFuse



Summary

Overview



Main concepts



High-Level Design



Tools



Code & visualization



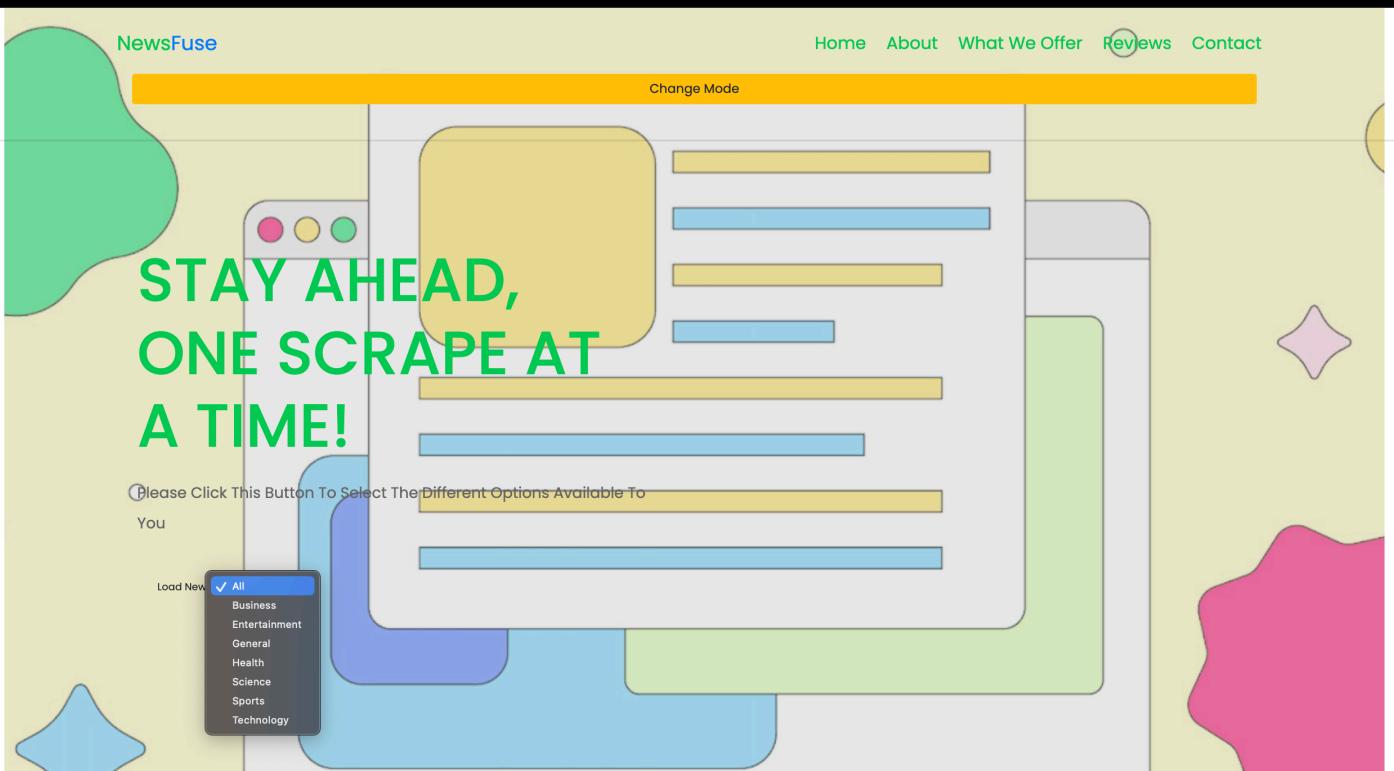
Yosr Jaouadi

Zeineb Belaid

Overview

NewsFuse is an innovative web scraper and news aggregation tool designed to collect, process, and present news articles providing users with a centralized platform to access the latest news grouped into various categories.

What We Achieved



Website interface (Safari)

The screenshot shows a list of news articles. The first article is titled "Bitcoin spot ETFs not the catalyst most thought it would be in 2024? - AMBCrypto News" with a sub-category of "business". It includes a snippet: "Grayscale added a negative outflow of over \$43M, triggering outflows for the other products. However, Bitcoin's market wasn't particularly affected by it." and a link to the full article. The second article is titled "Red, yellow, green ... and white? Smarter vehicles could mean big changes for the traffic light - The Associated Press" with a sub-category of "business". It includes a snippet: "As cars and trucks get smarter and more connected, the humble lights that have controlled the flow of traffic for more than a century could also be on the cusp of a major transformation. Researchers..." and a link to the full article. The third article is titled "Target announces Pride month merch will only be available in 'select stores' after last year's backlash - Fox Business" with a sub-category of "business". It includes a snippet: "After last year's backlash against some of Target's LGBTQIA+ products, the company announced their new Pride collection would only be available in "select stores." "We're offering a collection of pr..." and a link to the full article. The footer of the page says "Trump-Appointed USPS Postmaster General Draws Republican Rebuke - Newsweek".

Articles display (Chrome)

Main Concepts

Overview Components:

- Web Scraping

Functional Flow:

- Data Collection

- Data Aggregation
- News Categorization

Data Processing
Data Presentation

Pros & cons

Advantages

- **Customizability:** Tailored scraping to specific needs.
- **Freshness:** Real-time updates.
- **Ease of Use:** user-friendly, with intuitive interfaces and straightforward setup processes.
- **Cost-Efficiency:** Lower cost compared to subscription-based services.

- **Scalability:** Challenges in handling large-scale data.
- **Maintenance:** Regular updates are needed for scraping scripts.
- **Legal Issues:** Compliance with web scraping policies.
- **Layout Instability:** Websites frequently update their layout and structure, which can cause web scraping tools to fail if not regularly monitored and updated.

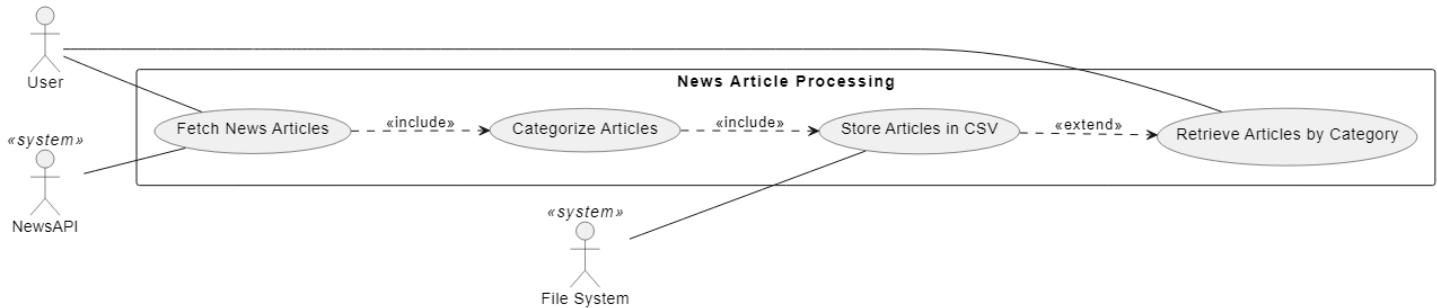
Limitations

High-level Design

Inputs & Outputs:

- Input: URLs, Keywords, Time Intervals
- Output: Aggregated News Data, categorized articles

Use Case Diagram:



Class Diagram:



Tools

- **Task:** Web Scraping
 - **Tools:** Python
- **Task:** Data Storage
 - **Tools:** JSON file
- **Task:** Front-end Development
 - **Tools:** HTML, CSS, JavaScript
 - **IDE:** Visual Studio Code

Development phases

1. Requirement Analysis

Define project goals, functionalities, and user requirements.
2. Design

Illustrates system architecture and components.
3. Implementation

Develop the web scraping engine to extract data from newsAPI website then store articles and display them according to the user preferences.
4. Testing

executes test cases to ensure the functionality, performance, and security of the web scrapping.
5. Deployment

Provide ongoing maintenance, updates, and support to ensure our NewsAggregator remains functional and up-to-date.
Also, address any user feedback or issues reported after deployment.

Code

The solution will consist of a web scraping tool built using Python, utilizing the **requests** library to scrape news articles from **newsAPI** website.

Executable Code

```
newsAPI.py 1  website.html  script.js  style.css  NewsFuse.py 1  newscraper.js
Users > yo > Desktop > NewsFuse > newsAPI.py > extract_article_info
1 import requests
2 import json
3
4 CATEGORIES = ['business', 'entertainment', 'general', 'health', 'science', 'sports', 'technology']
5
6 def fetch_articles(api_key, category):
7     url = f'https://newsapi.org/v2/top-headlines?country=us&category={category}&apiKey={api_key}'
8     response = requests.get(url)
9     if response.status_code == 200:
10         return response.json()['articles']
11     else:
12         print(f"Failed to fetch articles for category: {category}")
13         return None
14
15 def save_articles_to_json(articles, filename):
16     with open(filename, 'w') as json_file:
17         json.dump(articles, json_file, indent=4)
18
19 def extract_article_info(article, category):
20     return [
21         'title': article['title'],
22         'category': category,
23         'content': article['content'],
24         'url': article['url']
25     ]
26
27 def main():
28     api_key = '3991968999b844bab36533bc65292531'
29     all_articles = []
30     for category in CATEGORIES:
31         articles = fetch_articles(api_key, category)
32         if articles:
33             formatted_articles = [extract_article_info(article, category) for article in articles]
34             all_articles.extend(formatted_articles)
35
36     if all_articles:
37         save_articles_to_json(all_articles, 'articles.json')
38         print("Articles saved to articles.json")
39     else:
40         print("No articles fetched")
41
42 if __name__ == "__main__":
43     main()
```

Python

Articles are categorized into predefined topics: Business, Entertainment, Health, Science, Sports, Technology, and General.

This HTML structure forms the skeleton of the NewsFuse web page.

It includes a container (articles_container) where the news articles will be displayed and links to the CSS stylesheet (styles.css) and JavaScript file (script.js).

```
newsAPI.py 1   website.html  JS script.js  # style.css  NewsFuse.py 1  JS newspaper.js  newspaper.py
Users yo > Desktop > NewsFuse > website.html > html > body > header.header.fixed-top > div.container
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <link rel="stylesheet" href="/Users/yo/Desktop/NewsFuse/css/style.css" />
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>NewsFuse : Your quick updater</title>
9
10     <!-- font awesome cdn link -->
11     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
12
13     <!-- bootstrap cdn link -->
14     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/4.6.1/css/bootstrap.min.css">
15
16
17 </head>
18 <body>
19
20     <!-- header section starts -->
21
22     <header class="header fixed-top">
23
24         <div class="container">
25
26             <div class="row align-items-center justify-content-between">
27
28                 <a href="#" class="logo">NewsFuse</a>
29
30
31                 <nav class="nav">
32                     <a href="#">home</a>
33                     <a href="#">about</a>
34                     <a href="#">services</a>
35                     <a href="#">reviews</a>
36                     <a href="#">contact</a>
37
38             </nav>
39
40             <button class="btn btn-warning btn-lg btn-block m-4 p-2" onclick="toggleDarkMode()" id="changeModeBtn">
41                 Change Mode
42             </button>
43
44         </div>
45
46     </div>
47
48     <!-- header section ends -->
49
50     <!-- home section starts -->
51
52     <section class="home" id="home">
53
54         <div class="container">
55
56             <div class="row min-vh-100 align-items-center">
```

```
<html lang="en">
<body>
<section class="home" id="home">
    <div class="container">
        <div class="row min-vh-100 align-items-center">
            <div class="content text-center text-md-left">
                <h3>STAY AHEAD, One Scrape at a Time!</h3>
                <p>Please click this button to select the different options available to you </p>
                <div style="padding: 20px;">
                    <label for="category-select">Load News</label>
                    <select id="category-select">
                        <option value="all">All</option>
                        <option value="business">Business</option>
                        <option value="entertainment">Entertainment</option>
                        <option value="general">General</option>
                        <option value="health">Health</option>
                        <option value="science">Science</option>
                        <option value="sports">Sports</option>
                        <option value="technology">Technology</option>
                    </select>
                </div>
            </div>
        </div>
        <div id="articles-container" style="padding: 20px;"></div>
    </div>
</section>
<!-- home section ends -->
<!-- about section starts -->
<section class="about" id="about">
    <div class="container">
        <div class="row align-items-center">
            <div class="col-md-6 image">
                
            </div>
            <div class="col-md-6 content">
                <span>about us</span>
                <h3>From Data Streams to News Dreams!</h3>
                <p>Web scraping is a powerful technique used to extract data from websites. News aggregation involves the collection and categorization of news articles from TheG
```

HTML

Provides users with a web interface to browse articles by topic and access specific articles' categories.

```
newsAPI.py 1    < website.html      JS script.js X # style.css      NewsFuse.py 1      JS newscraper.js
Users > yo > Desktop > NewsFuse > JS script.js > ⚡ document.addEventListener('DOMContentLoaded') callback > ⚡ fetchArticles
1 // Function to toggle dark mode
2 function toggleDarkMode() {
3     document.body.classList.toggle('dark-mode');
4 }
5
6 // Function to fetch news articles from NewsAPI based on category
7 function fetchNewsFromAPI(category) {
8     const apiKey = '399196899b844bab3653bc65292531';
9     const url = `https://newsapi.org/v2/top-headlines?country=us&category=${category}&apiKey=${apiKey}`;
10
11     fetch(url)
12         .then(response => response.json())
13         .then(data => {
14             displayArticles(data.articles);
15         })
16         .catch(error => {
17             console.error('Error fetching news articles from NewsAPI:', error);
18         });
19 }
20
21 // Function to handle category selection
22 function handleCategorySelection(category) {
23     // Call the fetchNewsFromAPI function to fetch news articles from NewsAPI based on category
24     fetchNewsFromAPI(category);
25 }
26
27 // Function to display articles on the webpage
28 function displayArticles(articles) {
29     const newsContainer = document.getElementById('articles-container');
30     // Clear any existing content in the news container
31     newsContainer.innerHTML = '';
32
33     // Create HTML elements for each article
34     articles.forEach(article => {
35         const articleElement = document.createElement('div');
36         articleElement.innerHTML = `
37             <h3>${article.title}</h3>
38             <p>${article.description}</p>
39             <a href="${article.url}" target="_blank">Read More</a>
40         `;
41         newsContainer.appendChild(articleElement);
42     });
43 }
44
45 // Example: Fetch news articles for the default category (e.g., latest) when the page loads
46 document.addEventListener('DOMContentLoaded', function () {
47     const categorySelect = document.getElementById('category-select');
```

This JavaScript snippet fetches the news data

Desktop/NewsFuse/articles.json

endpoint,

parses the

JSON response,
and
dynamically

[www.wiley.com](#)

Appendix HTML

elements to

display the news articles on the

on the
webpage

```
sAPI.py 1   < website.html   JS script.js   ● # style.css   ⚡ NewsFuse.py 1   JS newspaper.js   ⚡ newspaper.py 1   ⚡ browser...  
yo > Desktop > NewsFuse > JS script.js ...  
function displayArticles(articles) {  
  articles.forEach(article => {  
    newsContainer.appendChild(articleElement);  
  });
}  
  
// Example: Fetch news articles for the default category (e.g., latest) when the page loads  
document.addEventListener('DOMContentLoaded', function () {  
  const categorySelect = document.getElementById('category-select');  
  const articlesContainer = document.getElementById('articles-container');  
  
  // Function to extract articles from JSON file  
  function fetchArticles() {  
    fetch('http://localhost:8000/articles.json')  
      .then(response => response.json())  
      .then(data => {  
        articlesContainer.innerHTML = ''; // Clear previous articles  
        const selectedCategory = categorySelect.value;  
        data.forEach(article => {  
          if ((selectedCategory === 'all' || article.category === selectedCategory) && article.content !== null && article.title !== '')  
          {  
            articlesContainer.innerHTML += `  
              <div class="article">  
                <h2>${article.title}</h2>  
                <p><strong>Category:</strong> ${article.category}</p>  
                <p>${article.content}</p>  
                <p>Access the full article by following this link : <a href="${article.url}" target="_blank">${article.url}</a></p>  
              </div>  
          };  
        });  
      })
      .catch(error => console.error('Error fetching articles:', error));  
  }
};  
  
// Event listener for category select change  
categorySelect.addEventListener('change', function() {  
  fetchArticles();  
});  
  
// Initial fetch for all articles  
fetchArticles();  
});
```

This is it ! A well-designed web scraper for a news aggregator combines technical expertise with ethical considerations. By understanding the theoretical aspects and implementing best practices, we created a reliable and efficient system for collecting news data.

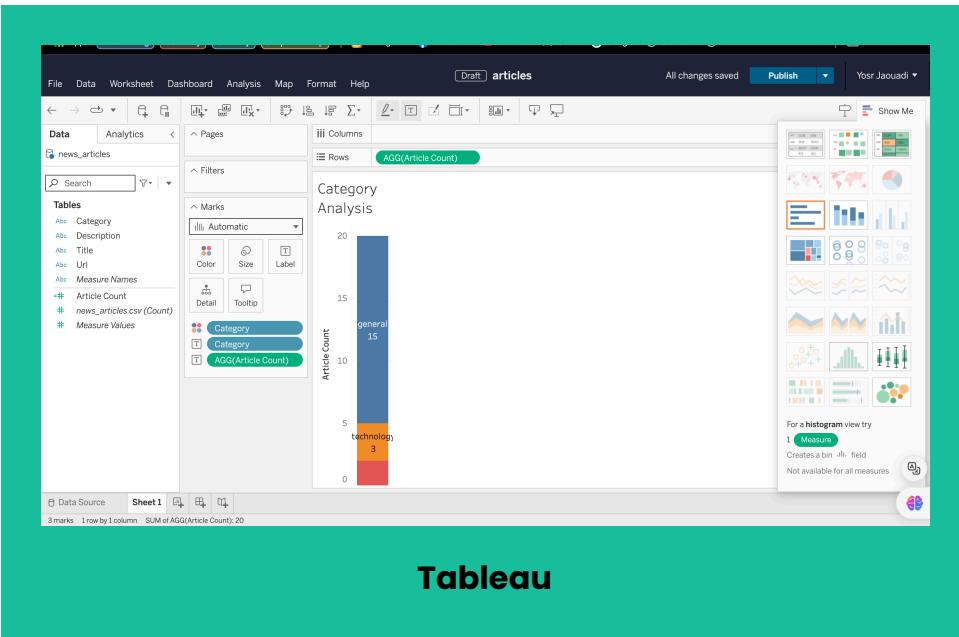
Code Output

A JSON file will be used each time to store the scraped data, displaying articles organized by title, description, URL, and category.

D28	A	B	C	D	E
1	title	description	url	category	
2	Biden Set to Imp (Bloomberg) -- P	https://finance.yahoo.com	politics		
3	California says n Service charges;	https://www.npr.org	general		
4	UN aid agency says Israelis set fir	https://www axios.com	general		
5	UK exits recessi Britain's econo	https://www.reut.com	business		
6	Pro-Palestine Sti Members of the	https://www.thecnn.com	politics		
7	Gaza war: Netar Israeli leader say	https://www.bbc.com	technology		
8	[Removed] [Removed]	https://removed.com	general		
9	Virginia school b School board me	https://www.cnn.com	general		
10	Senate passes b The Senate pass	https://www.npr.org	technology		
11	NBA playoffs: Lu P.J. Washington	https://sports.yahoo.com	general		
12	Jayson Tatum or Jayson Tatum sp	https://www.youtub.com	general		
13	Killing of an alim The fatal shootin	https://apnews.com	technology		
14	"Severe" solar storm could bring N	https://www axios.com	general		
15	Stormy Daniels trolls Trump after	https://thehill.com	general		
16	Justin Bieber's E Justin Bieber's e	https://www.eonit.com	technology		
17	ChatGPT maker OpenAI's search	https://nypost.com	technology		
18	Miss USA's resig The Miss USA w	https://www.nbc.com	technology		
19	M4 iPad Pro lacl One of the bigge	https://9to5mac.com	technology		
20	Cows have hum! Cows have the s	https://www.cnn.com	health		
21	What you misse Judge Juan Mer	https://www.nbc.com	politics		
22					
23					

Data Visualization

To provide a preliminary insight into the capabilities of NewsFuse, we conducted a trial visualization using **Tableau**.



1. We connected to our dataset stored in a CSV file.

-which includes news articles categorized under general news, weather, and technology.

2. we created a **category distribution visualization to analyze the proportion of news articles within these categories presented as a stacked bar chart.**

This trial demonstrates just a snippet of what category analysis can offer, highlighting how **Tableau** can effectively be used to uncover trends and patterns in our aggregated news data.

Potential Enhancements and Future Work

Advanced Data Visualization:

Implement graphs and charts to show news trends, popular topics, and article distribution using tools like Tableau.

User Analytics:

Analyze user interaction data to understand visitor demographics, preferences, and behavior.

Personalization:

Implement machine learning algorithms to personalize news feeds based on user interests and reading history.

Content Analysis:

Perform sentiment analysis on news articles to provide insights into the general tone of the news for handling content takedown requests and user data privacy.

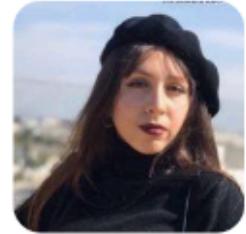
Contact Us

For further information, please reach out to us

Emails: jaouadi.yosr@tbs.u-tunis.tn & zaineb6belaid@gmail.com

Github:

Yoyo3333333/ NewsFuse



System Assurance and Security Project

0

Contributors

0

Issues

0

Stars

0

Forks

